Fermi National Accelerator Laboratory

FERMILAB-Conf-91/314

Performance and System Flexibility of the CDF Hardware Event Builder

T. Shaw and K. Schurecht

Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, Illinois 60510

P. Sinervo

University of Toronto, Dept. of Physics
60 St. George, Toronto, ON, Canada M5S 1A7

November 1991

* Presented at the *IEEE Nuclear Science Symposium*, Santa Fe, New Mexico, November 2-9, 1991.

Disclaimer

PERFORMANCE AND SYSTEM FLEXIBILITY OF THE CDF HARDWARE EVENT BUILDER

T. M. Shaw and K. Schurecht
Fermi National Accelerator Laboratory*
P. O. Box 500, Batavia, Il. 60510

P. K. Sinervo
University of Toronto
Dept. of Physics, 60 St. George, Toronto, ON, Canada M5S 1A7

ABSTRACT

The CDF Hardware Event Builder [1] is a flexible system which is built from a combination of three different 68020-based single width Fastbus modules. The system may contain as few as three boards or as many as fifteen, depending on the specific application. Functionally, the boards receive a command to read out the raw event data from a set of Fastbus based data buffers ("scanners"), reformat the data and then write the data to a Level 3 trigger/processing farm which will decide to throw the event away or to write it to tape. The data acquisition system at CDF will utilize two nine board systems which will allow an event rate of up to 35 Hz into the Level 3 trigger. This paper will present detailed performance factors, system and individual board architecture, and possible system configurations.

INTRODUCTION

The data acquisition system at Fermilab's Collider Detector Facility (CDF) consists of a large Fastbus network. Figure 1 gives an abbreviated overview of the network. The hardware Event Builder resides on this network and its job is to read out the front end scanners, reformat the raw data and ship the gathered and reformatted event to a Level 3 trigger/processor farm which will then decide whether the event will be written to tape or thrown away. Approximately 85 front end scanners, each with four internal buffers, must be read out by the Event Builder. These scanners are distributed among four Fastbus cable segments through which the Event Builder will read them out.

The Event Builder is directed in its actions by the Buffer Manager [2], a software process running on a VAX workstation which gains access to the Fastbus network through a Q-bus Processor Interface (QPI) module. The Buffer Manger provides centralized control and communication with the trigger system, Level 3 and the Event Builder. Event readout [3] is initiated when the Buffer Manager informs the Event Builder, through a Fastbus interrupt message, that an event is present in one of the front end buffers. The Event Builder will then initiatc a readout of the scanners. The readout from the four cable segments is performed simultaneously. When completed, the Event Builder informs the Buffer Manager. The Buffer Manager then waits for the Event Builder to inform it that it has reformatted the event into a YBOS [4,5] structure which will be recognized by Level 3. The Buffer Manager then tells the Event Builder to push the event to a given address in Level 3. The protocol for a single event concludes when the Event Builder informs the Buffer Manager that it has completed the push.

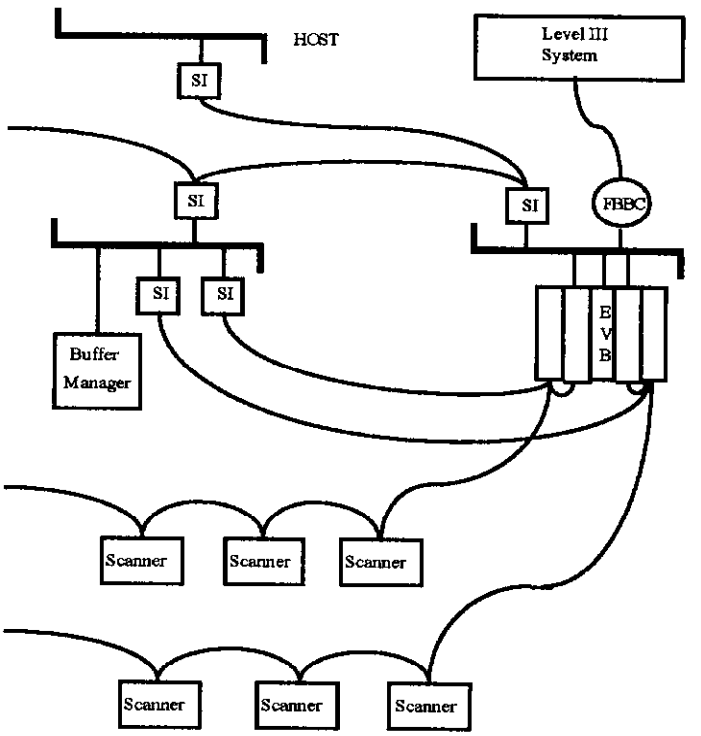

Fig. 1 Overview of CDF Data Acquisition Network

The maximum event processing rate of the Event Builder is defined as the rate at which the overall data acquisition system contributes no more than 5% overall deadtime (deadtime is accrued when all scanner buffers contain event data waiting to be read out.) The expected event size for the CDF detector is 180 kilobytes. Simulation results have predicted that a maximum event rate of 35 Hz will be achieved

* Operated by University Research Association, Inc., under contract with the U.S. Department of Energy

by installing two Event Builders in the CDF data acquisition network. The Buffer Manager would alternate event readout between the two Event Builders.

A minimal Event Builder system would consist of three boards - one Crate Controller [6], one Cable Controller, and one Reformatter. A maximum of fifteen boards can be installed in a single system. The Crate Controller communicates with the Buffer Manager and controls the push, or write, to Level 3. The Cable Controllers control the pull, or reading, of raw data from the scanners. The Reformatters are responsible for reformatting the raw event data. Boards communicate among themselves through a specially designed front panel bus. The proposed 9-board Event Builder systems which are planned for installation at CDF would consist of one Crate Controller, four Cable Controllers, and four Reformatters, see figure 2.

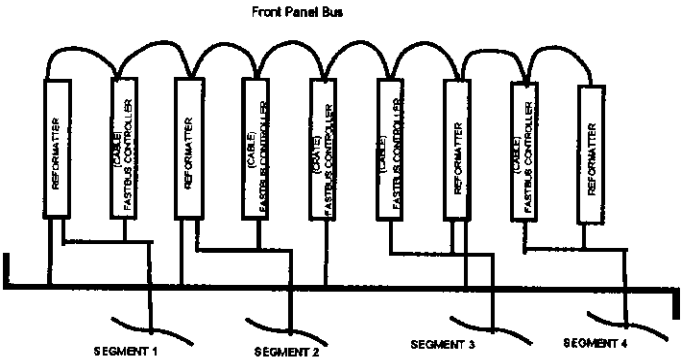

Fig. 2 Proposed CDF Event Builder Configuration

This paper will discuss the board and system level architecture, possible system configurations and performance of the Event Builder, and outline how we chose the system configuration best suited for CDF's application. The CDF terms of Buffer Manager and Level 3 will be used to help describe the Event Builder system, but, these terms really imply the need for some central task manager (Buffer Manager) and destination buffer (Level 3) for use with a generic data acquisition network.

EVENT BUILDER MODULES

The Event Builder consists of three different types of modules: a Crate Controller, a Cable Controller and a Reformatter. The Controller boards were based on the Aleph Event Builder, but their design was adapted to CDF's specific needs. The Reformatter board was essentially a new design.

All boards are based on Motorola's 68020 processor and run a version of Motorola's monitor software. Each of the Event Builder boards has 512 kilobytes of static RAM and 512 kilobytes of PROM. The modules also feature two RS-232 ports through which downloading and communication are possible.

A Front Panel bus was designed so that the boards could communicate among themselves without tying up valuable bandwidth on Fastbus. The bus physically consists of two 50 conductor cables which attach to the front panels of the Event Builder boards. Electrically, it uses RS-485 transceivers and supports up to 15 system boards.

THE CRATE CONTROLLER

The Crate Controller acts as the "traffic cop" for the Event Builder System. Only one Crate Controller is required in the Event Builder system. All communication between the Event Builder and the Buffer Manager is done through the Crate Controller. The Crate Controller is also the internal bookkeeper for the Event Builder. It keeps track of which of its internal buffers or "engines" are free, reformatting, or ready to be written out to Level 3. The Crate Controller also controls the writing of data to Level 3.

The Crate Controller has both Fastbus master and slave capabilities on the Fastbus crate segment. Its Fastbus interface is implemented as a coprocessor [7] to the MC68020. A microsequencer operating with an 88 bit field controls the actual gating of the Fastbus signals. A block diagram of the Crate Controller is found in Figure 3.

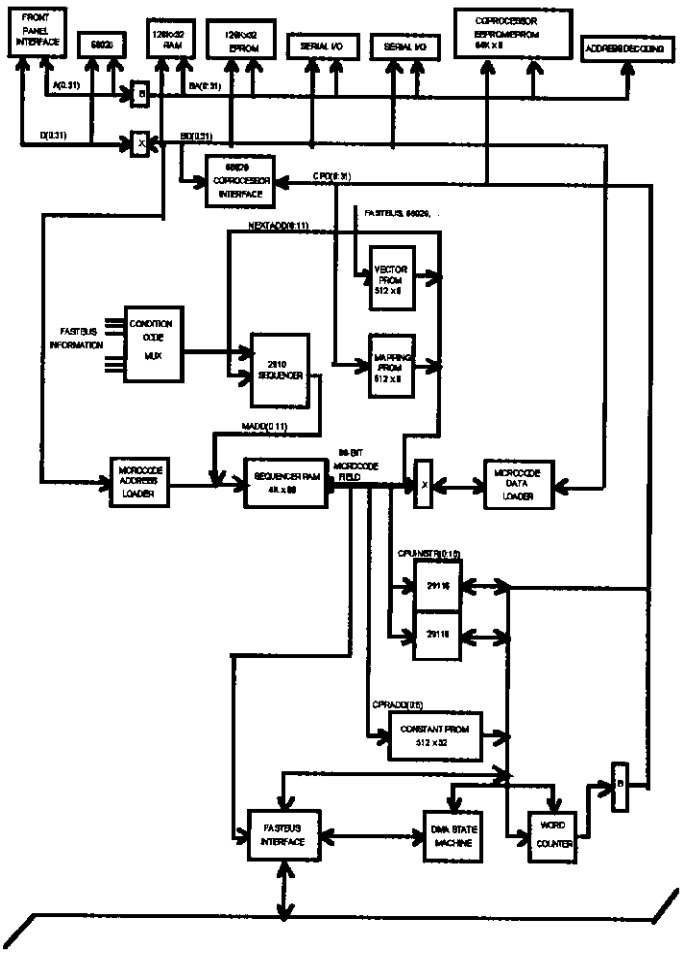

Fig. 3 Crate Controller Block Diagram

THE CABLE CONTROLLER

The Cable Controller has the primary responsibility of controlling the readout of the front-end scanners. The Cable Controller has the same Fastbus functionality as the Crate Controller, except that its electrical connection to Fastbus is to the Fastbus cable segment.

THE REFORMATTER

The Event Builder performs a DMA operation on the cable segment from each scanner to the Reformatter event buffer. The Cable Controller performs bus arbitration and the address cycles to select a given scanner and then toggles DS (data strobe) transitions on the cable segment to have the scanner place the data on the bus. The Reformatter spies on the bus and latches the data placed on the bus each time the scanner toggles DK (data acknowledge.)

The Reformatter itself can perform no Fastbus transactions, but it does have connections to the data and parity lines on both the Fastbus crate and cable segments. This allows it to spy on the segments and either pull data in or push it out on command from one of the Controller boards. The Reformatter reads data in from a Fastbus cable segment under command of the Cable Controller and writes data out under command of the Crate Controller onto the Fastbus crate segment.

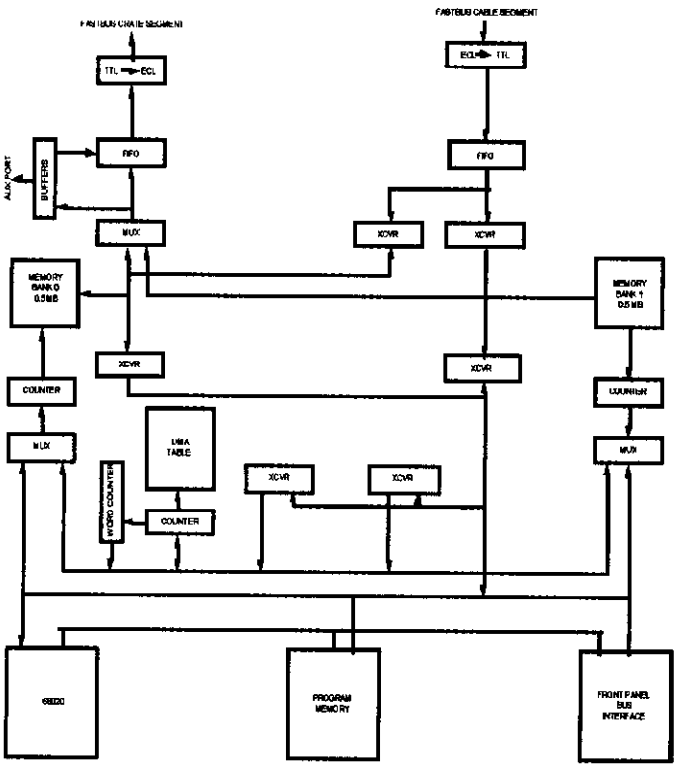

Fig. 4 Reformatter Block Diagram

Internally, the Reformatter has two "engines" or blocks of memory for receiving events; thus, it is capable of buffering two events at a time. After the data has been received by the Reformatter, the processor scans through the data and builds a DMA table that lists the order in which individual blocks of data are to be written out. This table includes the "header" blocks that identify the type of data and the length of the blocks. These header blocks are constructed by the processor as it scans the data. The arrangement of the data and header blocks in the DMA list ensures that the event information has been properly rearranged to group data from a common detector subsystem together, even though this data may have been read out from several scanners.

The design of the Reformatter allows it to simultaneously perform two of its three functions at any one time, i.e. reading from the front-end scanners, formatting or writing data to Level 3. Figure 4 shows a block diagram of the Reformatter board.

POSSIBLE SYSTEM CONFIGURATIONS

A maximum of fifteen boards may be present in an Event Builder system. One Crate Controller is required and from one to four Cable Controllers are to be included. Each Cable Controller requires at least one Reformatter, giving it two event buffers. Additional Reformatters could be added to work with a given Cable Controller if additional event buffers are required at the Event Builder level.

A minimum Event Builder system would consist of three boards: a Crate Controller, a Cable Controller, and a Reformatter. This would mean the the user's data acquisition system would be set up such that all front-end scanners are addressable and can be read out through a single Fastbus cable segment on which the Cable Controller and Reformatter would reside. Some task manager (in the case of CDF, this would be the Buffer Manager) would then communicate with the Crate Controller over the Fastbus crate segment and direct its actions. In addition, some higher level buffering device (Level 3 in the CDF system) would have to be addressable by the Crate Controller through the Fastbus crate segment so that the reformatted data could be dumped to it.

A larger example of an Event Builder system would include the following boards: one Crate Controller, four Cable Controllers and eight Reformatters. Thus, front-end scanners could be distributed along four separate Fastbus cable segments, each of which would be connected to one of the Cable Controllers. Also two Reformatters would be connected to each Fastbus cable segment to work with the resident Cable Controller.

Figure 2 illustrates the Fastbus crate and cable segment connections of a possible nine board configuration.

THE CDF EVENT BUILDER CONFIGURATION

THE ORIGINAL CDF CONFIGURATION

The original CDF configuration was a five board system which had one Crate Controller, two Cable Controllers and two Reformatters. Its throughput during the last CDF run was 4-7 Hz into Level 3, as documented in a previous paper [8]. Many of the possible board level, system level and software improvements which were described in that paper such as doubling the Front Panel Bus bandwidth, adding a second DMA table to the Reformatter, adding pipeline transfer capability, and improving the DMA mechanism were implemented in the new generation of Event Builder Boards that we are using now. However, it was anticipated that the original configuration would still be limited by an Event rate of 15-18 Hz. Since we wanted to squeeze the maximum performance out of the system, we began to investigate other possible system configurations.

CDF SYSTEM STUDIES AND PROPOSED CONFIGURATION

In order to determine the best system for CDF's application we began doing extensive timing measurements on an existing five board system. We also developed a Verilog behavioral model of the CDF data acquisition system. Detailed information about the form and results of this simulation is discussed in another paper being presented at this conference [9]. The simulation work was aided by the fact that we were able to measure the behavior of the components of the system. We found that the simulations accurately predicted the behavior of the five board system, and subsequently a nine board system. We therefore felt confident to let it analyze other configurations and began to be able to see the results of certain tradeoffs. For example, did it make sense to add more Reformatters to the cable segments or were the additional buffering they provided used so infrequently that the overhead in the Crate Controller for keeping track of them did not make it warranted?

Some of the timing parameters that form the input to the simulation are:

Reading Scanners - Pull Time
- ~2 ms in preparing for the read
- ~200 us setup per scanner to be read
- ~350 ns per word read

Reformatting of data
- ~5 ms overhead
- ~500 us per YBOS Bank built
- ~100 us per DMA pointer (1 pointer per component block in scanners)

Writing to Level 3 - Push Time
- ~4 ms preparing push
- ~2 ms overhead during push
- ~270 ns per word written

In addition, five to ten milliseconds are spent waiting for various messages for the buffer manager.

One of the first decisions to be made was to distribute the front-end scanners on four cable segments instead of the original two. Factors that influenced this distribution were the fact that certain data banks could not be split between cable segments and that the large overhead for reading out each scanner had to be factored in with the amount of data expected. In the initial system, the scanners were distributed based on the amount of data alone, which led to very unbalanced readout times on the two cable segments. A fairly balanced set of pull times was arrived at by taking these factors into account.

From that point, we decided to focus on four possible systems and use Verilog results as well as system tests to determine the best solution. Systems under study included a nine board system, shown in figure 2. The nine board system was now our minimum choice since there were now four cable segments to be read out. Also studied was a thirteen board system which would include an additional Reformatter on each Fastbus cable segment. The final question which we asked the simulator to resolve was whether or not it made sense to run two Event Builder systems with the Buffer Manager distributing events between them. The results of the simulations are reported below.

1 Event Builder 9 Board System	21 Hz
1 Event Builder 13 Board System	27 Hz
2 Event Builder 9 Board Systems	35 Hz
2 Event Builder 13 Board Systems	35 Hz

It was discovered that at 35 Hz, the Buffer Manager begins to be the bottleneck of the system. Therefore, a decision was made that the best solution for CDF was to run two 9 board Event Builder systems. Figure 5 illustrates how the connections to the scanners and Level 3 look to the Event Builder.

CONCLUSIONS

The Event Builder is a configurable system capable of reading data from front-end scanners which are distributed among one to four Fastbus cable segments and which writes formatted data out onto the Fastbus crate segment. A minimum system would include three boards, one Crate Controller, one Cable Controller and one Reformatter. Up to fifteen boards may be present in a system, which would have one Crate Controller, from one to four Cable Controllers and at least one Reformatter per Cable Controller.

System studies for the currently implemented data acquisition system at CDF have indicated that a 35 Hz throughput rate should be possible by running two nine board Event Builder systems. These systems have been installed and will be used in the upcoming CDF data run in 1992.

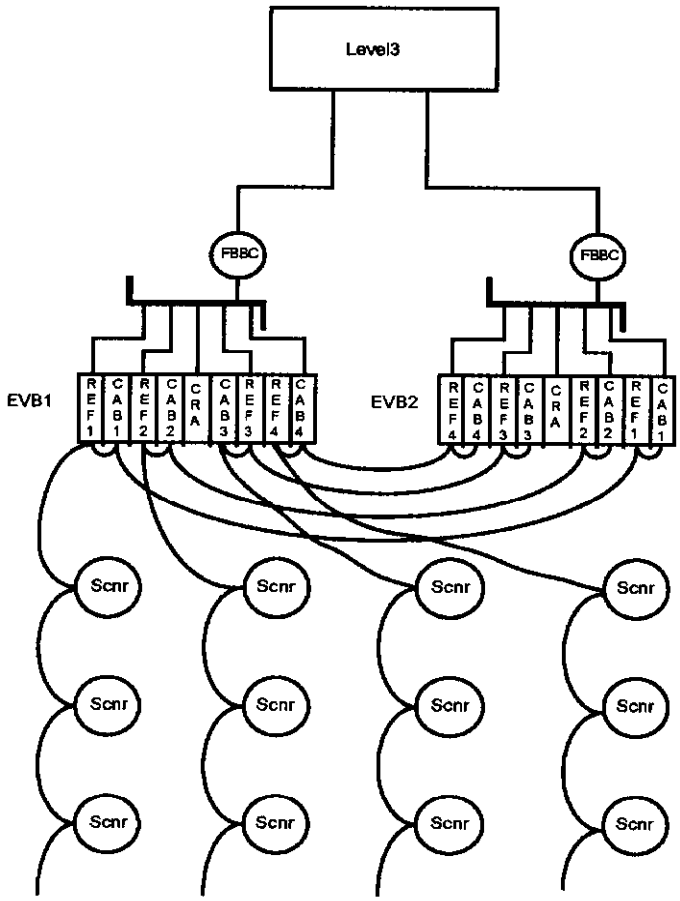

Fig. 5 Two Parallel Nine Board Event Builder Systems

REFERENCES

[1] A. W. Booth, M. Bowden and H. Gonzalez, "Specification for a Hardware Event Builder", CDF Note 452, Fermilab.

[2] C. Day, "Buffer Manager Software Design", CDF Note 326, Fermilab.

[3] P. K. Sinervo, et al, "Fast Data Acquisition with the CDF Event Builder", IEEE Transactions on Nuclear Science, Vol 36, No. 1, February 1989.

[4] J. T. Carroll, "YBOS Scanner Bank Format", CDF Note 264, Fermilab.

[5] D. Quarrie, et all, "CDF Event Structure", CDF Note 152, Fermilab.

[6] T. M. Shaw, "Hardware Description of the Controller Board", Event Builder Technical Note EVB/H/1, Fermilab.

[7] T. M. Shaw, "The Coprocessor Interface in the CDF Event Builder Technical Note EVB/H/4, Fermilab.

[8] T. M. Shaw, et al, "Architecture and Development of the CDF Hardware Event Builder", IEEE Transactions on Nuclear Science, Vol. 36, No. 1, February 1989.

[9] K. Schurecht, et al, "A Verilog Simulation of the CDF DAQ System", paper submitted to this symposium.